SCI INSTITUTE
TECHNICAL REPORT

Visualizing Unsteady Flows on Surfaces Using Spherical Parameterization

Guo-Shi Li, Xavier Tricoche, Charles Hansen

UUSCI-2007-013

Scientific Computing and Imaging Institute
University of Utah
Salt Lake City, UT 84112 USA

August 13, 2007

Abstract:

We present a view-independent, GPU-friendly method to visualize unsteady flows defined on surfaces using a global parameterization of the mesh. This is achieved by projecting the surface flow mapped to an intermediate spherical parameterization onto the faces of its bounding box. This transformation allows a consistent texture representation of a surface flow and avoids distracting artifacts typically observed in image-based approaches when changing the view parameters. Our method supports visualization techniques requiring the global view of the flow, and we demonstrate its application to GPUFLIC.

THE UNIVERSITY OF UTAH

Visualizing Unsteady Flows on Surfaces Using Spherical Parameterization

Guo-Shi Li, Xavier Tricoche, Charles Hansen

Abstract

We present a view-independent, GPU-friendly method to visualize unsteady flows defined on surfaces using a global parameterization of the mesh. This is achieved by projecting the surface flow mapped to an intermediate spherical parameterization onto the faces of its bounding box. This transformation allows a consistent texture representation of a surface flow and avoids distracting artifacts typically observed in image-based approaches when changing the view parameters. Our method supports visualization techniques requiring the global view of the flow, and we demonstrate its application to GPUFLIC.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Pictures/Image Generation

1. Introduction

The behavior of vector fields and flows defined over surfaces is of great interest to many engineering and biomedical disciplines with applications ranging from aeronautics to computational medicine. Texture representations offer a very intuitive means to display complex structure embedded in the flow. Pixels in the texture exhibit locally coherent intensity so that discernible patterns can form up over the entire domain. Image-based methods have received much attention from the visualization community in recent years due to the simple yet powerful solution they provide to display unsteady vector fields over curved surfaces [vW03,LJH03,WE04]. By projecting the flow attached to the visible part of the mesh to the image plane, image-space approaches can be applied to surfaces of arbitrary complexity and can be efficiently mapped to graphics hardware for interactive performance. However, processing only the visible part of the flow becomes a limiting factor if one attempts to further explore its non-local behavior, e.g. using Lagrangian techniques. One way to address this limitation is to extend image-space projection to cubemap space, in which the surface flow is projected to the six faces of the mesh bounding box. Yet, in the case of concave surfaces the flow typically cannot be fully projected because of self-occlusion. Furthermore, the projection introduces flow discontinuities along face boundaries in the cubemap space. To tackle these problems, we present a hybrid scheme combining the cubemap space projection and spherical parameterization which enables smooth and globally consistent representations of the flow on genus-zero surfaces. Our method can be efficiently mapped to graphics hardware and it supports texture-based flow visualization techniques not feasible in an image-space framework.

The rest of this paper is organized as follows. Section 2 discusses related work. Section 3 explains how a spherical parameterization can be used to create smooth and consistent surface flow representations in cubemap space. Section 4 illustrates how this scheme enables GPUFLIC [LTH06], a texture-based visualization method for planar vector fields, to visualize flows on genus-zero surfaces. Results are provided in section 5. We conclude with a discussion and suggest possible avenues for future work.

2. Related Work

For a detailed treatment of texture-based flow visualization techniques, please refer to [LHD*04] and the references therein. In this section we briefly review previous work most relevant to the method presented in the following, namely interactive dense flow visualization on polygonal meshes and surface parameterization.

An approach called *Image Based Flow Visualization* (IBFV) proposed by van Wijk [vW02] generates flow images by iteratively warping and blending noise textures. Its extension to surfaces (IBFVS), like Laramee et al.'s Image Space Advection (ISA), uses image-space computation to achieve interactive visualizations of flows defined over surfaces of arbitrary complexity [LvWJH04]. In both cases the flow covering the visible portion of the mesh is projected onto the image plane where texture advection takes place. Weiskopf and Ertl [WE04] presented a hybrid method capable of visualizing steady flows on surfaces. This is done by using the physical space flow on the visible part of the surface to perform per-pixel LIC operation with a 3D noise texture on the image plane.

There has been a tremendous amount of work devoted to mesh parameterization techniques. For a general overview, we refer readers to a comprehensive survey paper by Floater and Hormann [FH05]. Spherical parameterization schemes in particular, e.g. [PH03, GY03], have been the focus of numerous graphics publications in recent years. The simple shrink-wrap method described by Kobbelt et al. [KVLS99] turned out to be adequate for the datasets we tested. However more complex schemes can be used in the framework that we describe in the following sections to fit the needs of the considered application.

3. Cubemap Space Representation of Surface Flow

Our goal is to devise a scheme that converts the flow defined over a genus-zero mesh into a simple cubemap space parameterization of satisfactory quality to allow for a precise visual analysis. Each face of the cubemap space should contain the partial information about a portion of the surface flow, but also about its relationship with adjacent faces to permit a smooth flow advection across the artificial boundaries of the cube faces. Note that it requires that multiple faces contain different descriptions of the same region on the surface. In this case, the results obtained from different faces can be combined based on the characteristics of the flow representation on each face to yield the global image in object space.

Figure 1 illustrates the relationship between flow in physical space, spherical parameterization space, and the cubemap space under different conversion schemes. Straightforward projection cannot fully expand the surface flow from non-convex genus-zero meshes to the cubemap space because of self-occlusion (figure 1(a)). This problem can be solved by using any spherical parameterization scheme to convert the surface and the attached flow field into a sphere (figure 1(b)). After the conversion, one may attempt to project the flow now defined on the sphere onto the cube faces along radial direction. This method evenly distributes the surface flow to faces of the cubemap space, in which the distortion in the flow representation increases as one moves away from the center of the cube faces (figure 1(c)). The strongest distortion happens along face boundaries. This can

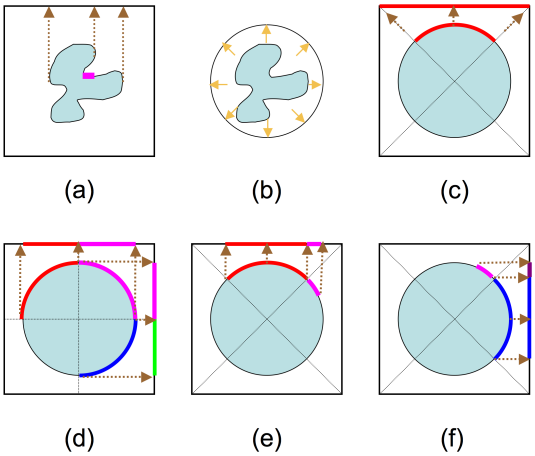

Figure 1: *Mapping surface flow to cubemap space. (a) Self-occlusion with non-convex object . (b) Spherical parameterization. (c) Radial projection – increasing distortion toward the boundary. (d) Orthogonal projection – adjacent half circle (red and blue) overlap each other (purple). (e) Proposed restricted orthogonal projection of surface (red) with buffer zone (purple) overlapping adjacent face. (f) Another example of projection of the surface (blue) on the face adjacent to the one in (e).*

negatively impact operations performed on the cubemap representation such as flow texture synthesis, which may lead to visually distracting artifacts. Another option is to perform orthogonal projection to non-occluded faces (figure 1(d)). In 2D, each face receives exactly half of the surface flow with little or mild distortion except in the direct vicinity of the boundary. Although the region subjected to strong distortion is relatively small on each face, it can significantly reduce the overall resulting visual quality since the flow representation on every face in this projection scheme is entirely overlapped with those on adjacent ones.

In contrast, the projection strategy we propose is illustrated in figure 1(e) and (f). For each face, we orthogonally project a subset of the mesh onto the closest cube face. In 2D, this subset is defined by the end points of diagonals. This scheme alleviates the boundary distortion issue. The downside of this approach is that the flow on the surface becomes artificially discontinuous beyond the "cut". This is problematic for various numerical schemes commonly used in flow visualization, which typically require a smooth underlying flow. To mitigate this problem, we attach a small *buffer zone*, surrounding the surface to be projected called *central zone*. The flow projected from the buffer zone has higher distortion. However, it is less likely to impact the overall quality of the subsequent flow visualization since it is only used as a

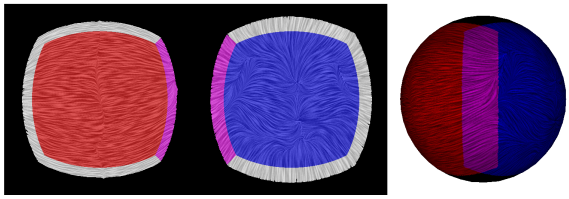

Figure 2: *Center zone and buffer zone in cubemap space and physical space. Left: texture on cube face corresponding to figure 1(e). Middle: texture corresponding to figure 1(f). Right: In physical space.*

transition between adjacent central zones and its size is kept small. Please refer to Figure 2.

4. GPUFLIC on Surface Flow

The scheme described in section 3 can be used in combination with many flow visualizations, especially texture-based techniques requiring global representation of the surface flow. In this section, we illustrate the use of the cubemap space representation by adapting GPUFLIC [LTH06] to the visualization of surface flow.

Designed for planar vector fields, GPUFLIC [LTH06] is a texture-based method that visualizes unsteady flows by advecting a dense set of particles forward in time while creating coherent patterns by depositing their color attributes along the path that they follow. Because of the very nature of this method a global representation is necessary to enable this algorithm to function on surface flows. With the cubemap space flow representation and a pre-computed spherical parameterization, one can generate the flow texture in the cubemap space, and then visualize the texture in the physical space by standard texture mapping with the spherical parameterization as the texture coordinates. Areas simultaneously covered by multiple faces are blended together. We also utilize the buffer zone to smoothly transport particles across adjacent faces. When a particle enters the buffer zone, it is redirected to the corresponding location on another face based on the overlapping configuration of the surface regions. Please refer to figures 1(e) and (f) and 2. The cubemap space flow can also be generated on the GPU, as the projection scheme described in section 3 can be efficiently executed by the graphics hardware. The conversion of the flow field from the physical space to the cubemap space is done by using the projected spherical parameterization to convert the coordinates frame from the local tangent plane to the cubemap space using the fragment shader.

5. Results

We implemented the algorithm described in section 4 using C++ and OpenGL. Two datasets, `Sphere` and `Heart` are used in our experiments. `Sphere` is a 3D numerical simulation in which a viscous fluid is flowing around a spherical obstacle. The surface flow corresponds to the shear stress

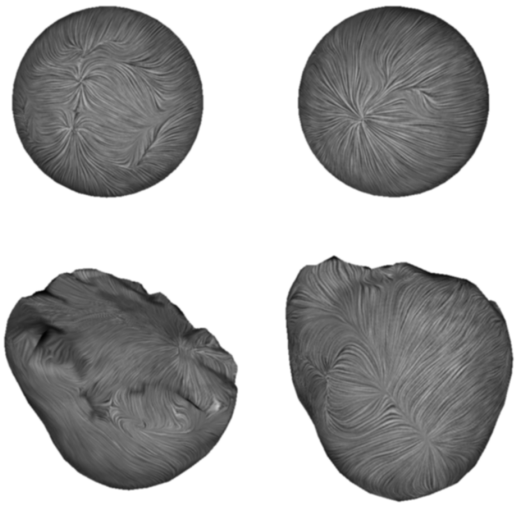

Figure 3: *GPUFLIC results. Top: Sphere. Bottom: Heart. Please refer to the color plate for more images.*

vector field. `Heart` is the electric field on the cardiac surface extracted from a numerical computation based on experimental data. Please refer to figure 3. Table 1 shows the performance matrix on a standard Windows PC equipped with dual nVIDIA 7800 GTX cards configured in SLI mode. The spherical parameterization of the all datasets takes less than one minute on a Pentium 4 3.0GHz machine with 2 GB of RAM, and is computed as preprocessing.

6. Conclusion and Future Work

We have described the cubemap space representation of the surface flow using the spherical parameterization. This scheme can be used as a basic framework for techniques requiring a global view of the vector field, such as GPUFLIC or dye advection on surface flows. Admittedly, the image-space approach of flow texture synthesis such as [LJH03] and [vW03] provides even higher frame rates and is relatively easy to implement. The contribution of this paper, however, is to provide a view-independent flow representation, which enables consistent patterns on the surface to be visualized.

This method provides many avenues of further investigation. The spherical parameterization limits the type of geometry applicable to our approach. The distortion in the vector field induced by the parameterization, and its relationship to the visual quality in the visualization needs to be addressed in a more sophisticated way. We would like to explore other flow analysis/visualization application (such as dye advection or topology analysis) and to explore other parameterization schemes for flows on different type of meshes.

dataset	Sphere		Heart	
resolution	512x768	1024x1536	512x768	1024x1536
life span = 1	49.6	3.1	49.2	2.8
life span = 2	9.6	1.4	8.3	1.1

Table 1: *Performance in FPS. The resolution denotes the size of the computation space (in pixels), and life span is the active duration of the particles.*

References

[FH05] FLOATER M. S., HORMANN K.: Surface parameterization: a tutorial and survey. In *Advances in Multiresolution for Geometric Modelling*, Dodgson N. A., Floater M. S., Sabin M. A., (Eds.), Mathematics and Visualization. Springer, Berlin, Heidelberg, 2005, pp. 157–186.

[GY03] GU X., YAU S.-T.: Global conformal surface parameterization. In *SGP '03: Proceedings of the 2003 Eurographics/ACM SIGGRAPH symposium on Geometry processing* (Aire-la-Ville, Switzerland, Switzerland, 2003), Eurographics Association, pp. 127–137.

[KVLS99] KOBBELT L. P., VORSATZ J., LABSIK U., SEIDEL H.-P.: A shrink wrapping approach to remeshing polygonal surface. In *Computer Graphics Forum (Eurographics '99)* (1999), Brunet P., Scopigno R., (Eds.), vol. 18(3), The Eurographics Association and Blackwell Publishers, pp. 119–130.

[LHD$^?$ 04] LARAMEE R., HAUSER H., DOLEISCH H., VROLIJK B., POST F., WEISKOPF D.: The state of the art in visualization: Dense and texture-based techniques. *Computer Graphics Forum 23*, 2 (2004), 143–161.

[LJH03] LARAMEE R. S., JOBARD B., HAUSER H.: Image space based visualization of unsteady flow on surfaces. In *Proceedings of the 14th IEEE Visualization 2003* (Washington, DC, USA, 2003), IEEE Computer Society, p. 18.

[LTH06] LI G.-S., TRICOCHE X., HANSEN C.: Gpuflic: Interactive and accurate dense visualization of unsteady flows. In *Eurographics/IEEE-VGTC Symposium on Visualizations* (2006).

[LvWJH04] LARAMEE R. S., VAN WIJK J. J., JOBARD B., HAUSER H.: Isa and ibfvs: Image space-based visualization of flow on surfaces. *IEEE Transactions on Visualization and Computer Graphics 10*, 6 (2004), 637–648.

[PH03] PRAUN E., HOPPE H.: Spherical parametrization and remeshing. *ACM Trans. Graph. 22*, 3 (2003), 340–349.

[vW02] VAN WIJK J.: Image based flow visualization. In *Proceedings of the 29th annual conference on Computer graphics and interactive techniques* (2002), ACM Press, pp. 745–754.

[vW03] VAN WIJK J. J.: Image based flow visualization for curved surfaces. In *Proceedings of the 14th IEEE Visualization 2003* (Washington, DC, USA, 2003), IEEE Computer Society, p. 17.

[WE04] WEISKOPF D., ERTL T.: A hybrid physical/device-space approach for spatio-temporally coherent interactive texture advection on curved surfaces. In *Proceedings of the 2004 conference on Graphics interface* (School of Computer Science, University of Waterloo, Waterloo, Ontario, Canada, 2004), Canadian Human-Computer Communications Society, pp. 263–270.

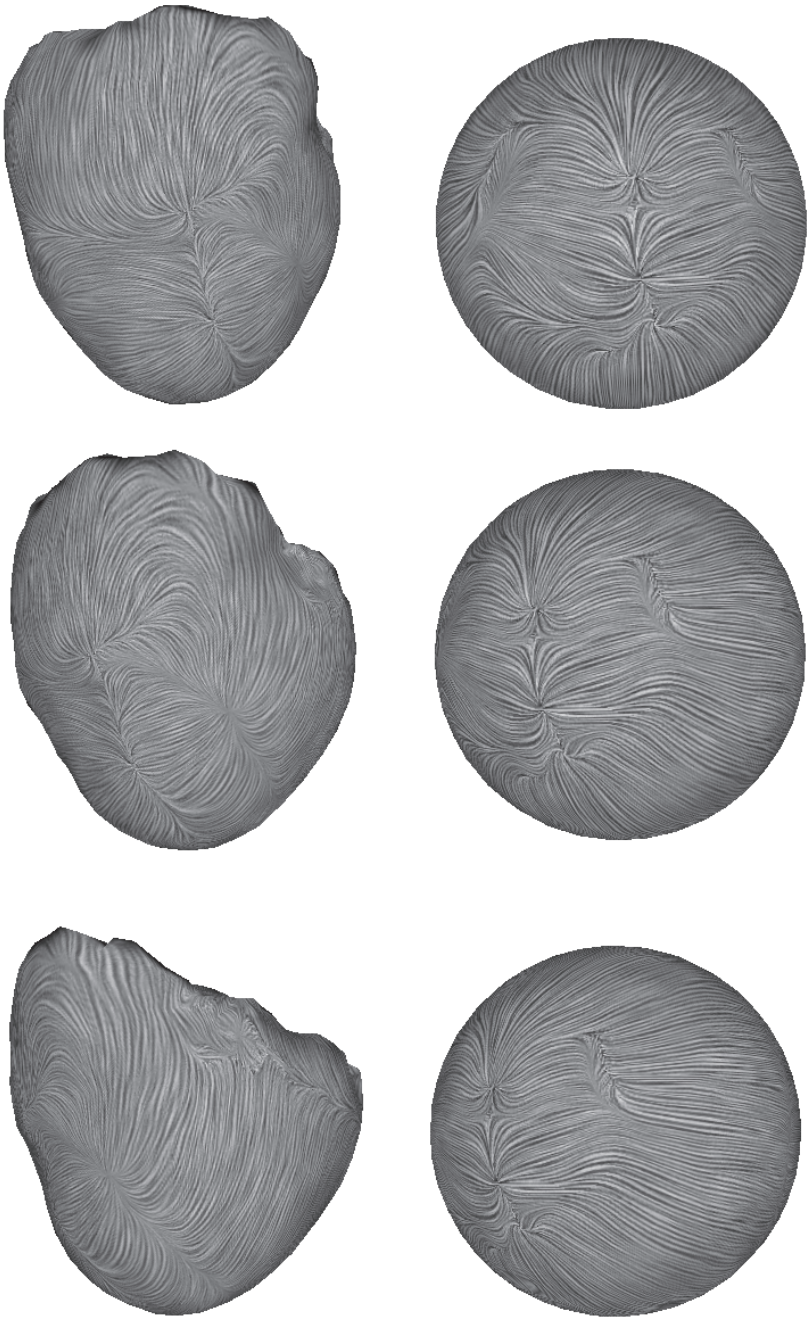

Figure 4: *Color Plates - GPUFLIC results. Left: Sphere. Right: Heart. Note that the texture patterns are consistent on adjacent views.*